Individuality transfer: Predicting human decision-making across task conditions

Hiroshi Higashi[1]

[1]Graduate School of Engineering, The University of Osaka

October 23, 2025

Abstract

Predicting an individual's behaviour in one task condition based on their behaviour in a different condition is a key challenge in modeling individual decision-making tendencies. We propose a novel framework that addresses this challenge by leveraging neural networks and introducing a concept we term the "individual latent representation." This representation, extracted from behaviour in a "source" task condition via an encoder network, captures an individual's unique decision-making tendencies. A decoder network then utilizes this representation to generate the weights of a task-specific neural network (a "task solver"), which predicts the individual's behaviour in a "target" task condition. We demonstrate the effectiveness of our approach in two distinct decision-making tasks: a value-guided task and a perceptual task. Our framework offers a robust and generalizable approach for parameterizing individual variability, providing a promising pathway toward computational modeling at the individual level—replicating individuals in silico.

1 Introduction

Humans (and other animals) exhibit substantial commonalities in their decision-making processes. However, considerable variability is also frequently observed in how individuals perform perceptual and cognitive decision-making tasks [5, 1]. This variability arises from differences in underlying cognitive mechanisms. For example, individuals may vary in their ability or tendency to retain past experiences [12, 8], respond to events with both speed and accuracy [52, 45], or explore novel actions [17]. If these factors can be meaningfully disentangled, they would enable a concise characterization of individual decision-making processes, yielding a low-dimensional, parameterized representation of individuality. Such a representation could, in turn, be leveraged to predict future behaviours at an individual level. Shifting from population-level predictions to an individual-based approach would mark a significant advancement in domains where precise

behaviour prediction is essential, such as social and cognitive sciences. Beyond prediction, this approach offers a framework for parameterizing and clustering individuals, thereby facilitating the visualization of behavioural heterogeneity, which has applications in psychiatric analysis [31, 10]. Furthermore, this parameterization offers a promising pathway toward computational modeling at the individual level—replicating the cognitive and functional characteristics of individuals *in silico* [41].

Cognitive modelling is a standard approach for reproducing and predicting human behaviour [29, 3, 57], often implemented within a reinforcement learning framework (e.g., [30, 9, 54]). However, because these cognitive models are manually designed by researchers, their ability to accurately fit behavioural data may be limited [16, 44, 28, 13]. A data-driven approach using artificial neural networks (ANNs) offers an alternative [11, 33, 39]. Unlike cognitive models, which rely on predefined behavioural assumptions [37], ANNs require minimal prior assumptions and can learn complex patterns directly from data. For instance, convolutional neural networks (CNNs) have successfully replicated human choices and reaction times in various visual tasks [25, 35, 15]. Similarly, recurrent neural networks (RNNs) [42, 7] have been applied to model value-guided decision-making tasks such as the multi-armed bandit problem [56, 10]. A promising approach to capturing individual decision-making tendencies while preserving behavioural consistency is to tune ANN weights using a parameterized representation of individuality.

This idea was first proposed by Dezfouli et al. [10], who employed an RNN to solve a two-armed bandit task. Their study utilized an autoencoder framework [38, 48], in which behavioural recordings from a single session of the bandit task, performed by an individual, were fed into an encoder. The encoder produced a low-dimensional vector, interpreted as a latent representation of the individual. Similar to hypernetworks [20, 22], a decoder then took this low-dimensional vector as input and generated the weights of the RNN. This framework successfully reproduced behavioural recordings from other sessions of the same bandit task while preserving individual characteristics. However, since this individuality transfer has only been validated within the bandit task, it remains unclear whether the extracted latent representation captures an individual's intrinsic tendencies across a variety of task conditions.

To address this question, we aim to make the low-dimensional representation—referred to as the *individual latent representation*—robust to variations across individuals and task conditions, thereby enhancing its generalizability. Specifically, we propose a framework that predicts an individual's behaviours, not only in the same condition but also in similar yet distinct task conditions and environments. If the individual latent representation serves as a low-dimensional representation of an individual's decision-making process, then extracting it from one condition could facilitate the prediction of that individual's behaviours in another.

In this study, we define the problem of *individuality transfer across task conditions* as follows (also illustrated in Figure 1). We assume access to a behavioural dataset from multiple individuals performing two task conditions: a

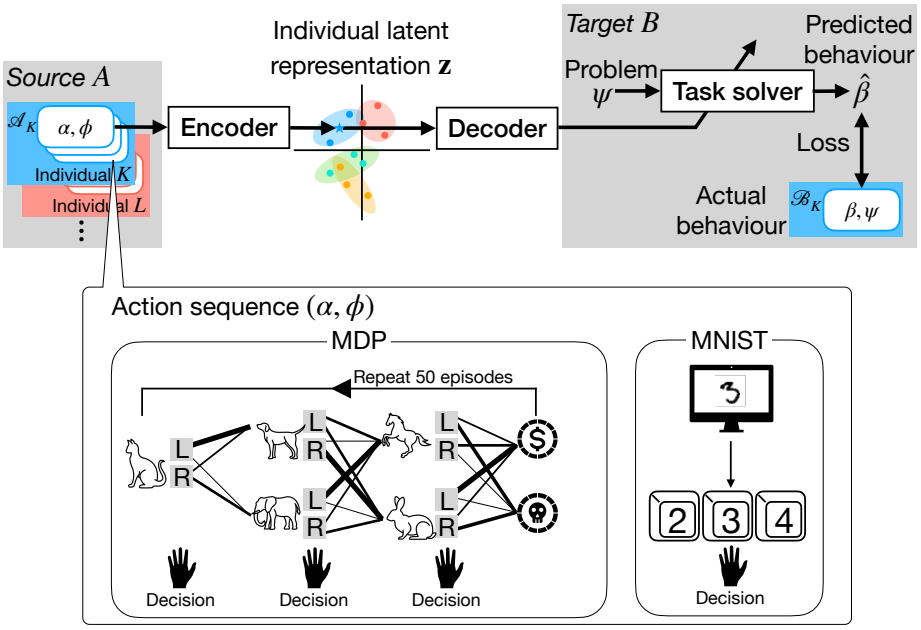


Figure 1: The EIDT (encoder, individual latent representation, decoder, and task solver) framework for individuality transfer across task conditions. The encoder maps action(s) α, provided by an individual K performing a specific problem ϕ in the source task condition A, into an individual latent representation (represented as a point in the two-dimensional space in the center). The individual latent representation is then fed into the decoder, which generates the weights for a task solver. The task solver predicts the behaviour of the same individual K in the target task condition B. During the training, a loss function evaluates the discrepancy between the predicted behaviour $\hat{\beta}$ and the actual recorded behaviour β of individual K. The encoder's input is referred to as an *action sequence*, the form of which depends on task. For example, in a sequential Markov decision process (MDP) task, an action sequence consists of an environment (state transition probabilities) and a sequence of actions over multiple episodes. For a digit recognition task, it consists of a stimulus digit image and the corresponding chosen response.

source task condition and a *target task condition*. We train an encoder that takes behavioural data from the source task condition as input and outputs an individual latent representation. This representation is then fed into a decoder, which generates the weights of an ANN, referred to as a *task solver*, that reproduces behaviours in the target task condition. For testing, a new individual provides behavioural data from the source task condition, allowing us to infer his/her individual latent representation. Using this representation, a task solver is constructed to predict how the test individual will behave in the target task condition. Importantly, this prediction does not require any behavioural data from the test individual performing the target task condition. We refer to this framework as *EIDT*, an acronym for encoder, individual latent representation, decoder and task solver.

We evaluated whether the proposed EIDT framework can effectively transfer individuality in both value-guided sequential decision-making tasks and perceptual decision-making tasks. To assess its generalizability across individuals, meaning its ability to predict the behaviour of previously unseen individuals, we tested the framework using a test participant pool that was not included in the dataset used for model training. To determine how well our framework captures each individual's unique behavioural patterns, we compared the prediction performance of a task solver specifically designed for a given individual with the performance of task solvers designed for other individuals. Our results indicate that the proposed framework successfully mimics decision-making while accounting for individual differences.

2 Results

We evaluated our EIDT framework using two distinct experimental paradigms: a value-guided sequential decision-making task (MDP task) and a perceptual decision-making task (MNIST task). For each paradigm, we assessed model performance in two scenarios. The first, *Within-Condition Prediction*, tested a model's ability to predict behaviour within a single task condition without individuality transfer. In this scenario, a model was trained on data from a pool of participants to predict the behaviour of a held-out individual in that same condition. The second, *Cross-Condition Transfer*, tested the core hypothesis of individuality transfer. Here, a model used behavioural data from a participant in "source" condition to predict that same participant's behaviour in a different "target" condition.

The prediction performance was evaluated using two metrics: the negative log-likelihood on a trial-by-trial basis, and the rate for behaviour matched. The negative log-likelihood is based on the probability the model assigned to the specific action that the human participant actually took on that trial. The rate for behaviour matched measures the proportion of trials where the model's most likely action (deterministically predicted by sampling from the output probabilities) matched the participant's actual choice.

2.1 Markov decision process (MDP) task

The dataset consisted of behavioural data from 81 participants who performed both 2-step and 3-step MDP tasks. Each participant completed three blocks of 50 episodes for each condition, resulting in 486 action sequences in total. All analyses were performed using a leave-one-participant-out cross-validation procedure. For each fold, the model was trained on 80 participants, with 90% used for training updates and 10% for validation-based early stopping.

Task solver accurately predicts average behaviour. First, we validated our core neural network architecture in Within-Condition Prediction. We trained a standard task solver, using the architecture defined in Section 4.2.4, on the training/validation pool ($N = 80$) to predict the behaviour of the held-out participant. We compared its performance against a standard cognitive model (a Q-leaning model, Section 4.2.3) whose parameters were averaged from fits to the same training/validation pool.

As shown in Figure 2, the neural network-based task solver significantly outperformed the cognitive model. A two-way (model: cognitive model/task solver, task condition: 2-step/3-step) repeated-measures (RM) ANOVA with Greenhouse-Geisser correction (significant level was 0.05) revealed a significant effect of the model on both negative log-likelihood (model: $F_{1,80} = 148.828$, $p < 0.001$, $\eta^2_G = 0.143$, task condition: $F_{1,80} = 1.107$, $p = 0.296$, $\eta^2_G = 0.002$, interaction: $F_{1,80} = 0.240$, $p = 0.626$, $\eta^2_G < 0.001$) and the rate for behaviour matched (model: $F_{1,80} = 110.684$, $p < 0.001$, $\eta^2_G = 0.165$, task condition: $F_{1,80} = 3.914$, $p = 0.051$, $\eta^2_G = 0.009$, interaction: $F_{1,80} = 19.059$, $p < 0.001$, $\eta^2_G = 0.014$). This result confirms that our RNN-based architecture serves as a strong foundation for modeling decision-making in this task.

EIDT enables accurate individuality transfer. Next, we tested our main hypothesis in Cross-Condition Transfer. We used the full EIDT framework to predict a participant's behaviour in a target condition (e.g., 3-step MDP) using their behavioural data from a source condition (e.g., 2-step MDP). We compared the performance of two models:

Cognitive model A Q-learning model whose parameters (q_{lr}, q_{init}, q_{dr}, and q_{it}) were individually fitted for each participant using their data from the source condition and then applied to predict behaviour in the target condition.

EIDT Our framework, trained on the training and validation pool using data from both source and target conditions (see Figure S2, Supplementary Materials for representative training and validation curves). To predict behaviour for a test participant, their individual latent representation was computed by averaging the encoder's output across all of their behavioural sequences from the source condition, and this representation was fed to the decoder to generate the task solver weights. For reference, the averaged in-

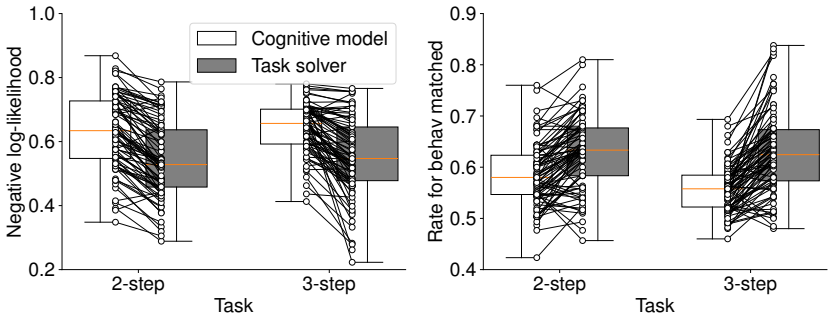


Figure 2: Comparison of prediction performance in Within-Condition Prediction for the MDP task. The plots show the negative log-likelihood (left) and the rate for behaviour matched (right) for the average-participant cognitive model and the task solver for 2-step and 3-step conditions. Box plots indicate the median and interquartile range. Whiskers extend to the minimum and maximum values. Each connected pair of dots represents a single participant's data. The task solver demonstrates significantly better performance.

dividual latent representations are visualized in Figure S3, Supplementary Materials.

The EIDT framework demonstrated significantly better prediction accuracy than the individualized cognitive model (Figure 3). A two-way (model: cognitive model/EIDT, transfer direction: 2→3/3→2) RM ANOVA confirmed a significant effect of the model on negative log-likelihood (model: $F_{1,80} = 95.705$, $p < 0.001$, $\eta^2_G = 0.142$, transfer direction: $F_{1,80} = 14.255$, $p < 0.001$, $\eta^2_G = 0.019$, interaction: $F_{1,80} = 0.008$, $p = 0.012$, $\eta^2_G = 0.002$) and the rate for behaviour matched (model: $F_{1,80} = 100.843$, $p < 0.001$, $\eta^2_G = 0.132$, transfer direction: $F_{1,80} = 13.021$, $p = 0.001$, $\eta^2_G = 0.011$, interaction: $F_{1,80} = 0.964$, $p = 0.329$, $\eta^2_G < 0.001$). This result indicates that EIDT successfully captures and transfers individual-specific behavioural patterns more effectively than a traditional parameter-based transfer approach.

Latent space distance predicts transfer performance. To verify that the individual latent representation meaningfully captures individuality, we conducted a "cross-individual" analysis. We generated a task solver using the latent representation of one participant (Participant l) and used it to predict the behaviour of another participant (Participant k). We then measured the relationship between the prediction performance ($y_{k,l}$) and the Euclidean distance ($d_{k,l}$) between the latent representations of Participants k and l.

As hypothesized, prediction performance was strongly dependent on this distance (Figure 4). We fitted the data using a generalized linear model (GLM): $y_{k,l} \sim \text{Gamma}\left(\log(\beta_{\text{participant}_k} + \beta_d d_{k,l} + \beta_0)\right)$. The fitting confirmed that distance ($d_{k,l}$) was a significant predictor: the coefficient β_d was significantly pos-

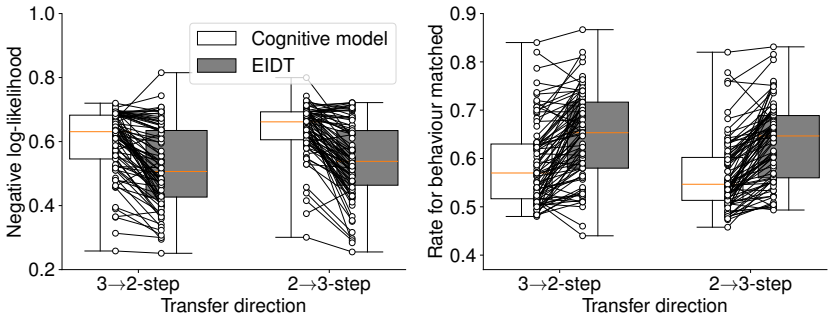


Figure 3: Individuality transfer performance in Cross-Condition Transfer for the MDP task. The plots compare the EIDT framework against an individualized cognitive model on negative log-likelihood (left) and rate for behaviour matched (right) for both 2-step to 3-step and 3-step to 2-step transfer. Box plots indicate the median and interquartile range. Whiskers extend to the minimum and maximum values. Each connected pair of dots represents a single participant's data. The EIDT model shows superior prediction accuracy.

itive for negative log-likelihood (transfer direction 3→2: $\beta_d = 0.176$, $p < 0.001$, 2→3: $\beta_d = 0.316$, $p < 0.001$) and significantly negative for the rate for behaviour matched (3→2: $\beta_d = 0.106$, $p < 0.001$, 2→3: $\beta_d = -0.149$, $p < 0.001$). This indicates that prediction performance degrades as the behavioural dissimilarity (represented by distance in the latent space) between the source and target individual increases, providing direct evidence that the latent space organizes individuals by behavioural similarity.

On-policy simulations generate human-like behaviour. To assess if our model could generate realistic behaviour, we conducted on-policy simulations. Task solvers specialized to each individual via EIDT performed the MDP task using the same environments as the human participants. We compared the model behaviour to human behaviour on two metrics: total reward per block and the rate of highly-rewarding action selected in the final step.

The model-generated behaviours closely mirrored human behaviours (Figure 5). We found significant correlations between humans and their corresponding models in both total rewards (3→2: $R = 0.667$, $p < 0.001$; 2→3: $R = 0.593$, $p < 0.001$) and the rate of highly-rewarding action selected (3→2: $R = 0.889$, $p < 0.001$; 2→3: $R = 0.835$, $p < 0.001$). This demonstrates that the EIDT framework captures individual tendencies that generalize to active, sequential behaviour generation.

Individual latent representations reflect cognitive parameters. To better interpret the latent space, we applied our EIDT model (trained only on human data) to simulated data from 1,000 Q-learning agents. The agents had known learning rates (q_{lr}) and inverse temperatures (q_{it}) sampled from

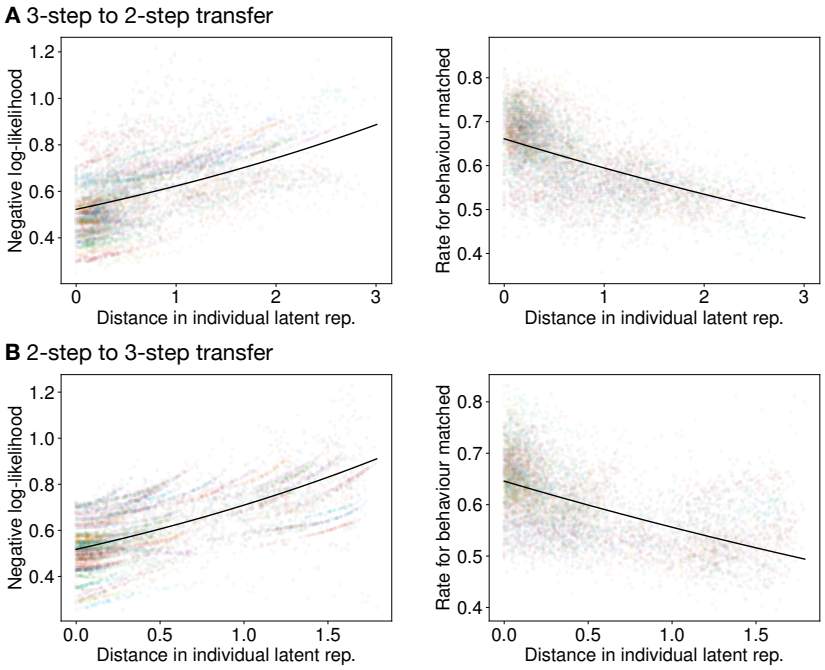


Figure 4: Prediction performances as a functions of latent space distance in the MDP task. This cross-individual analysis shows the result of using a task solver generated from one participant to predict the behaviour of another participant. The horizontal axis is the Euclidean distance between the latent representation of the two participants. The vertical axis shows the negative log-likelihood (left) and rate for behaviour matched (right). Each dot represents one participant pair. Performance degrades as the distance between individuals increases, with the solid line showing the GLM fit. **A** 3-step to 2-step transfer. **B** 2-step to 3-step transfer.

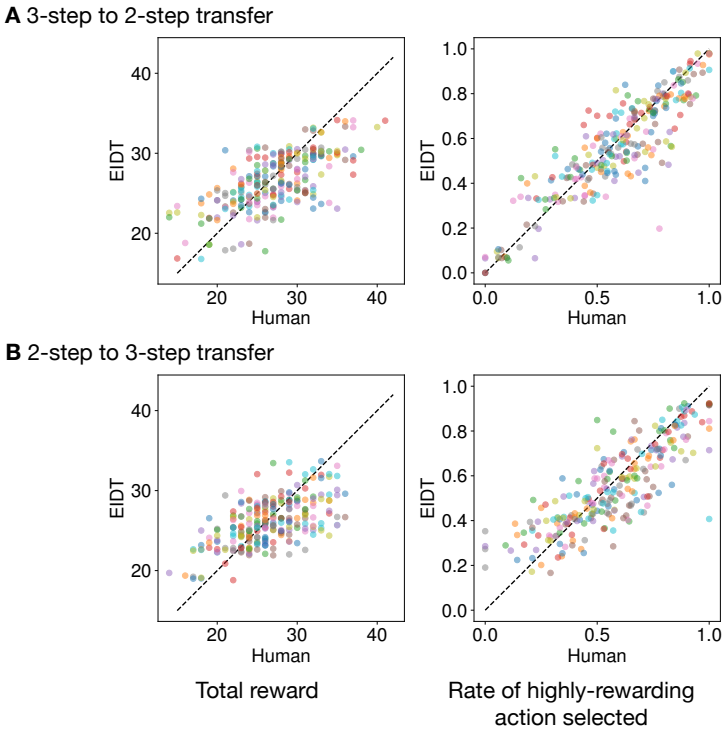


Figure 5: Comparison of on-policy behaviour between humans and EIDT-generated task solvers. Each dot represents the performance of a single human participant (horizontal axis) versus their corresponding model (vertical axis) for one block. Plots show the total reward (left) and the rate of highly-rewarding action selected (right). **A** 3-step to 2-step transfer. **B** 2-step to 3-step transfer.

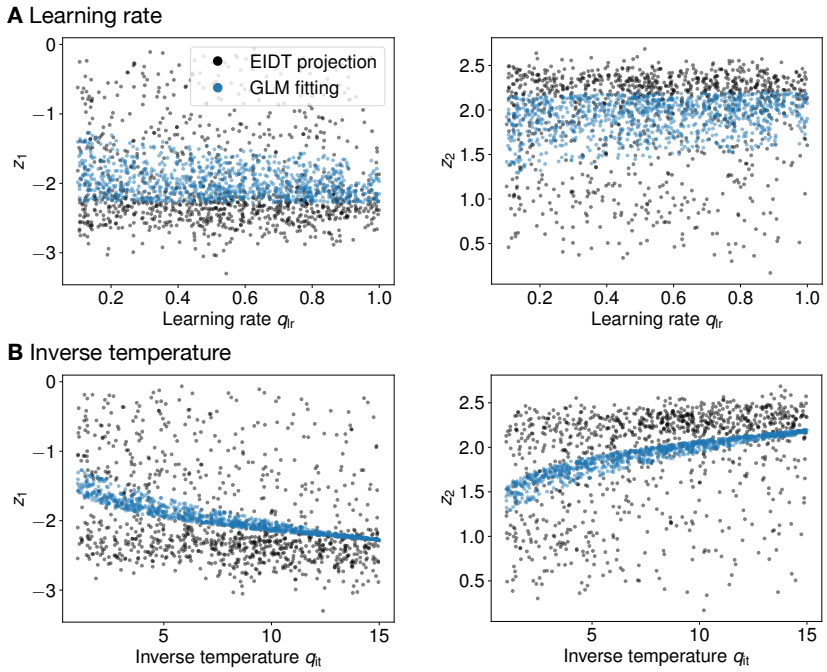


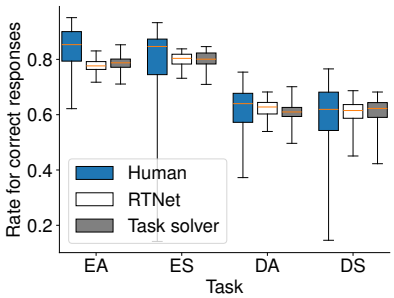
Figure 6: Mapping of Q-learning parameters to the individual latent space for the 3-step MDP task. Each plot shows one dimension of the latent representation (z_1 (left) or z_2 (right)) as a function of either the learning rate (q_{lr}, **A**) or the inverse temperature (q_{it}, **B**) of simulated Q-learning agents. Black dots represent the latent representation produced by the encoder from the agent's behaviour. Blue dots show the fit from a GLM.

distributions matched to human fits (Figure S1, Supplementary Materials). A cross-individual analysis on these agents confirmed that latent space distance predicted performance, mirroring the results from human data (Figure S5, Supplementary Materials).

The results revealed a systematic mapping between the cognitive parameters and the coordinates of the individual latent representation (Figures 6 and S4, Supplementary Materials). A GLM analysis (Table S1, Supplementary Materials) showed that both both q_{rl} and q_{it} (and their interaction) were significant predictors of the latent dimensions (z_1 and z_2). This indicates that our data-driven representation captures core computational properties defined in classic reinforcement learning theory.

2.2 Handwritten digit recognition (MNIST) task

We then sought to replicate our findings in a different domain: perceptual decision-making. We used data from Rafiei et al. [34], where 60 participants



Task condition	Difficulty	Focus
EA	Easy	Accuracy
ES	Easy	Speed
DA	Difficult	Accuracy
DS	Difficult	Speed

Figure 7: Task performance (rate of correct responses) in Within-Condition Prediction for the MNIST tasks. Box plots indicate the median and interquartile range. Whiskers extend to the minimum and maximum values. Performance is compared across human participants, the RTNet model, and our task solver for the four experimental conditions (EA, ES, DA, and DS). All three show similar performance patterns.

identified noisy images of digits under four conditions varying in difficulty and speed-accuracy focus (EA: easy, accuracy focus, ES: easy, speed focus, DA: difficult, accuracy focus, and DS: difficult, speed focus). Analyses were again conducted using leave-one-participant-out cross-validation.

Task solver outperforms RTNet. First, in Within-Condition Prediction, our base task solver demonstrated task performance (rate of correct responses indicating how accurately a human participant or model responded to the stimulus digit) comparable to human participants and established RTNet model [34] (Figure 7). A two-way (model: human/RTNet/Task solver, task condition: EA/ES/DA/DS) RM ANOVA showed no significant effect of model type ($F_{2,118} = 1.546$, $p = 0.219$, $\eta^2_G = 0.008$), while the task condition had a significant effect ($F_{3,177} = 866.322$, $p < 0.001$, $\eta^2_G = 0.684$). This confirms similar task-solving ability.

However, the task solver significantly outperformed RTNet in predicting participants' trial-by-trial choices (Figure 8). A two-way RM ANOVA revealed significant effects of on both negative log-likelihood (model: $F_{1,59} = 1312.328$, $p < 0.001$, $\eta^2_G = 0.731$, task condition: $F_{3,177} = 460.535$, $p < 0.001$, $\eta^2_G = 0.682$, their interaction: $F_{3,177} = 24.476$, $p < 0.001$, $\eta^2_G = 0.026$) and the rate for behaviour matched (model: $F_{1,59} = 43.544$, $p < 0.001$, $\eta^2_G = 0.005$, task condition: $F_{3,177} = 455.728$, $p < 0.001$, $\eta^2_G = 0.701$, their interaction: $F_{3,177} = 11.052$, $p < 0.001$, $\eta^2_G = 0.002$). This confirms the task solver's suitability for modeling individual behaviour in this task.

EIDT accurately transfers individuality. Next, in Cross-Condition Transfer, we tested individuality transfer across all 12 pairs of experimental conditions. The full EIDT framework was compared againt a baseline: a task solver

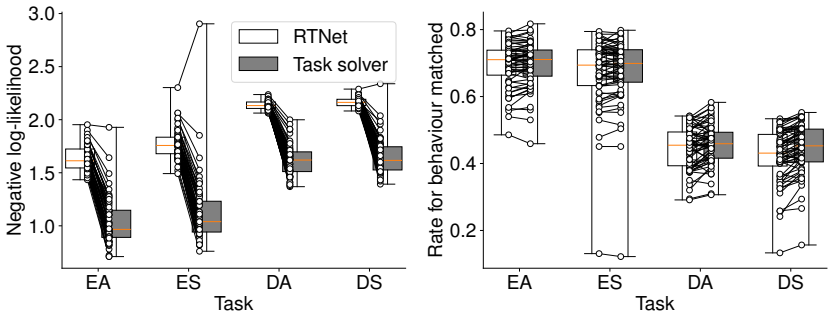


Figure 8: Comparison of prediction performance in Within-Condition Prediction for the MNIST task. The plots show the negative log-likelihood (left) and the rate for behaviour matched (right) for the RTNet model and our task solver. Each connected pair of dots represents a single participant's data. Box plots indicate the median and interquartile range. Whiskers extend to the minimum and maximum values. The task solver achieves significantly better prediction accuracy.

(source) model trained directly on a test participant's source condition data.

The EIDT framework consistently and significantly outperformed this baseline across all transfer sets (Figure 9). A two-way (model: task solver/EIDT, transfer direction: 12 sets (see horizontal axis)) RM ANOVA confirmed a significant effect of the model on negative log-likelihood (model: $F_{3,177} = 2440.373$, $p < 0.001$, $\eta_G^2 = 0.800$, transfer direction: $F_{11,649} = 347.850$, $p < 0.001$, $\eta_G^2 = 0.616$, interaction: $F_{33,1947} = 336.968$, $p < 0.001$, $\eta_G^2 = 0.573$) and rate for behaviour matched (model: $F_{3,177} = 2318.456$, $p < 0.001$, $\eta_G^2 = 0.798$, transfer direction: $F_{11,649} = 394.753$, $p < 0.001$, $\eta_G^2 = 0.591$, interaction: $F_{33,1947} = 355.577$, $p < 0.001$, $\eta_G^2 = 0.628$). The model was also able to reproduce idiosyncratic error patterns of individual participants, such as Participant #23's lower accuracy for digit 1 and Participant #56's difficulty with digits 6 and 7 (Figure 10).

Latent space reflects behavioural tendencies. Similar to the MDP task, a cross-individual analysis showed that the distance in the latent space was significant predictor of prediction performance for all transfer directions (Figure 11; see Figures S8, S9, and Table S2, Supplementary Materials, for full results). This confirms that, in the perceptual domain as well, the individual latent representation captures meaningful behavioural differences that are critical for accurate prediction.

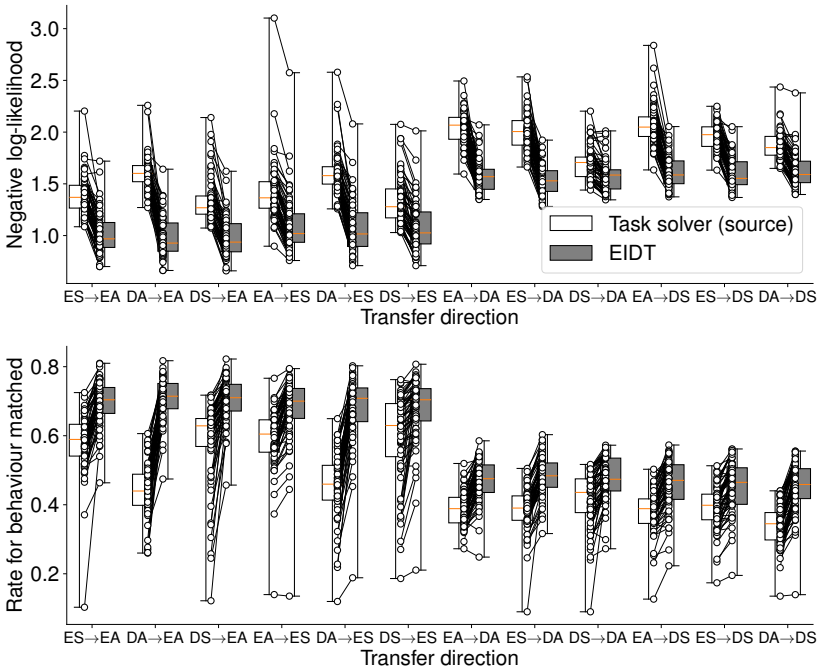


Figure 9: Individuality transfer performance in Cross-Condition Transfer for the MNIST task. The plots compare the EIDT framework against the task solver (source) baseline across all 12 transfer directions on negative log-likelihood (top) and rate for behaviour matched (bottom). Each connected pair of dots represents a single participant's data. Box plots indicate the median and interquartile range. Whiskers extend to the minimum and maximum values. EIDT consistently demonstrates superior prediction accuracy.

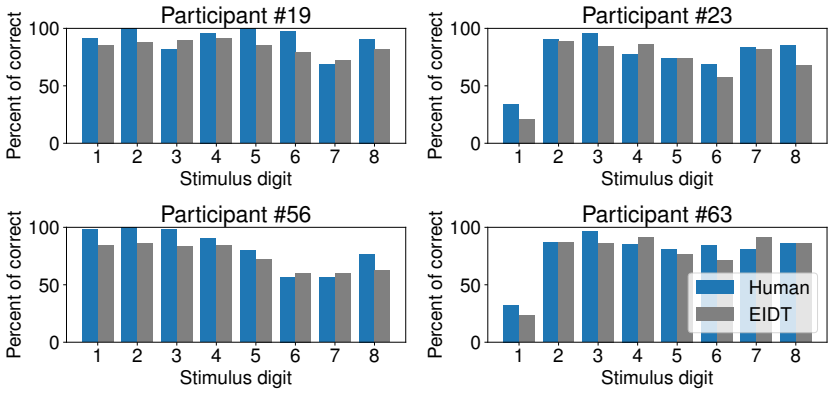


Figure 10: EIDT captures individual-specific error patterns in the MNIST task. The plots show the percentage of correct responses for each digit for four representative participants (blue bars) and their corresponding EIDT-generated models (gray bars). Data shown is for the ES target condition, with transfer from EA.

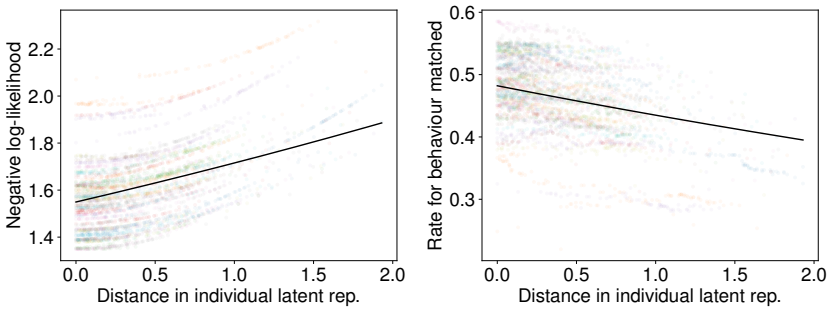


Figure 11: Prediction performance as a function of latent space distance in the MNIST task (transfer direction EA→DA). This cross-individual analysis shows the result of using a task solver generated from one participant to predict the behaviour of another participant. The horizontal axis is the Euclidean distance between the latent representation of the two participants. The vertical axis shows the negative log-likelihood (left) and rate for behaviour matched (right). Each dot represents one participant pair. Performance degrades as the distance between individuals increases, with the solid line showing the GLM fit.

3 Discussion

We proposed an EIDT framework for modeling the unique decision-making process of each individual. This framework enables the transfer of an individual latent representation from a (source) task condition to a different (target) task condition, allowing a task solver predict behaviours in the target task condition. Several neural network techniques, such as autoencoders [38, 48], hypernetworks [20], and learning-to-learn [53, 43], facilitate this transfer. Our experiments, conducted on both value-guided sequential and perceptual decision-making tasks, demonstrated the potential of the proposed framework in individuality transfer across task conditions.

EIDT framework extends prior work on individuality transfer. The core concept of using an encoder-decoder architecture to capture individuality builds on the work of Dezfouli et al. [10], who applied a similar model to a bandit task. We extended this idea in three key ways. First, we validated that the framework is effective for previously unseen individuals who were not included in model training. Although these individuals provided behavioural data in the source task condition to identify their individual latent representations, their data were not used for model training. Second, we established that this transfer is effective across different experimental conditions (e.g., changes in task rules or difficulty), not just across sessions of the same task. Third, while the original work focused on value-guided tasks, we validated the framework's applicability to perceptual decision-making tasks, specifically the MNIST task. These findings establish that EIDT effectively captures individual differences across both task conditions and individuals.

Interpreting the individual latent representation remains challenging. Although we found that Q-learning parameters were reflected in the individual latent representation, the interpretation of this representation remains an open question. Since interpretation often requires task-condition-specific considerations [13], it falls outside the primary scope of this study, whose aim is to develop a general framework for individuality transfer. Previous research [28, 18] has explored associating neural network parameters with cognitive or functional meanings. Approaches such as disentangling techniques [2] and cognitive model integration [19, 49, 44, 14] could aid in better understanding the cognitive and functional significance of the individual latent representation.

Regarding the individual latent representation, disentanglement and separation losses [10] during the model training could enhance interpretability. However, we used only the reproduction loss, as defined in (5), because interpretable parameters in cognitive models (e.g., [9]) are not necessarily independent (e.g., an individual with a high learning rate may also have a high inverse temperature [27], resulting these two parameters being represented with one variable).

Why can the encoder extract individuality for unseen individuals? Our experiments, which divided participants into training and test participant pools, demonstrated that the framework successfully extracts individuality for completely new individuals. This generalization likely relies on the fact that individuals with similar behavioural patterns result in similar individual latent representation and individuals similar to new participants exist in the training participant pool [57]. This hypothesis suggests that individuals can be clustered based on behavioural patterns. Behavioural clustering has been widely discussed in relation to psychiatric conditions, medication effects, and gender-based differences (e.g., [31, 50, 40]). Our results could contribute to a deeper discussion of behavioural characteristics by clustering not only these groups but also healthy controls.

Which processes contribute to individuality? In the MNIST task, we assumed that individuality emerged primarily from the decision-making process (implemented by an RNN [45, 6]), rather than from the visual processing system (implemented by a CNN [55]). The CNN was pretrained, and the decoder did not tune its weights. Our results do not rule out the possibility that the visual system also exhibits individuality [24, 47]; however, they imply that individual differences in perceptual decision-making can be explained primarily by variations in the decision-making system [36, 51, 57, 21]. This assumption provides valuable insights for research on human perception.

Limitations. One limitation is that the source and target behaviours were performed on different conditions, but within the same task. Thus, our findings do not fully evaluate the generalizability of individuality transfer across diverse task domains. However, our framework has the potential to be applied to diverse tasks since it connects the source and target tasks via the individual latent representation and accepts completely different tasks for the source and target. A key to realizing this transfer might be ensuring that the cognitive functions, such as memory, required for solving the source and target tasks are (partially) shared. The latent representation is expected to represent individual features of these functions. Conversely, if source and target tasks require completely different functions to solve them, the transfer by EIDT would not work.

The effectiveness of individuality transfer may be influenced by dataset volume. As discussed earlier, prediction performance may depend on whether similar individuals exist in the training participant pool. In our study, 100 participants were sufficient for effective transfer. However, tasks involving greater behavioural diversity may require a substantially larger dataset.

As discussed earlier, the interpretability of the individual latent representation requires further investigation. Furthermore, the optimal dimensionality of the individual latent representation remains unclear. This likely depends on the complexity of tasks involved—specifically, the number of factors needed to represent the diversity of behaviour observed in those tasks. While these factors have been explored in cognitive modeling research (e.g., [23, 13]), a clear

understanding at the individual level is still lacking. Integrating cognitive modeling with data-driven neural network approaches [10, 19] could help identify key factors underlying individual differences in decision-making.

Future directions. To further generalize our framework, a large-scale dataset is necessary, as discussed in the limitations. This dataset should include a large number of participants to ensure prediction performance for diverse individuals [32]. All participants should perform the same set of tasks, which should include a variety of tasks [56]. Building upon our framework, where the encoder currently accepts action sequences from only a single task, a more generalizable encoder should be able to process behavioural data from multiple tasks to generate a more robust individual latent representation. To enhance the encoder, a multi-head neural network architecture [4] could be utilized. A individual latent representation would enable transfer to a wider variety of tasks and allow accurate and detailed parameterization of individuals using data from only a single task.

Robust and generalizable parameterization of individuality enables computational modeling at the individual level. This approach, in turn, makes it possible to replicate individuals' cognitive and functional characteristics *in silico* [41]. We anticipate that it offers a promising pathway toward a new frontier: artificial intelligence endowed with individuality.

4 Methods

4.1 General framework for individuality transfer across task conditions

We formulate the problem of individuality transfer, which involves extracting an individual latent representation from a source task condition and predicting behaviour in a target task condition with preserving individuality. We consider two task conditions, A and B, which are different but related. For example, condition A might be a 2-step MDP, while condition B is a 3-step MDP.

The individuality transfer across task conditions is defined as follows. An individual K performs a problem within condition A, with their behaviour recorded as $\mathcal{A}_K$. Our objective is to predict $\mathcal{B}_K$, which represents K's behaviour when performing a task with condition B. To achieve this, we extract an individual latent representation $\boldsymbol{z}$ from $\mathcal{A}_K$, capturing the individual's behavioural characteristics. This representation $\boldsymbol{z}$ is then used to construct a task solver, enabling it to mimic K's behaviour in condition B. Since condition A provides data for estimating the individual latent representation and condition B is the target of behaviour prediction, we refer to them as the source task condition and target task condition, respectively.

Our proposed framework for the individuality transfer consists of three modules:

Task solver predicts behaviour in the target condition B.

Encoder extracts the individual latent representation from the source condition A.

Decoder generates the weights of the task solver based on the individual latent representation.

These modules are illustrated in Figure 1. We refer to this framework as EIDT, an acronym for encoder, individual latent representation, decoder and task solver.

4.1.1 Data representation

For training, we assume that behaviour data from a participant pool $\mathcal{P}$ ($K \notin \mathcal{P}$), where each participant has performed both conditions A and B. These datasets are represented as $\mathcal{A} = \{\mathcal{A}_n\}_{n \in \mathcal{P}}$ and $\mathcal{B} = \{\mathcal{B}_n\}_{n \in \mathcal{P}}$.

For each individual n, the set $\mathcal{A}_n$ consists of one or more sets, each containing a problem instance ϕ (stimuli, task settings, or environment in condition A) and a sequence of action(s) α (recorded behavioural responses). For example, in an MDP task, ϕ represents the Markov process (state-action-reward transition) and α consists of choices over multiple trials. In a simple object recognition task, ϕ is a visual stimulus and α is the participant's response to the stimulus. Similarly, $\mathcal{B}_n$ consists of a problem instance ψ and an action sequence β.

4.1.2 Task solver

The task solver predicts the action sequence for condition B as

$$\hat{\beta} = \mathrm{TS}(\psi; \Theta_{\mathrm{TS}}), \tag{1}$$

where ψ is a specific problem in condition B and Θ_{TS} represents the solver's weights. The task solver architecture is tailored to condition B. For example, in an MDP task, the task solver outputs a sequence of actions in response to ψ. In a simple object recognition task, it produces an action based on a visual stimulus ψ.

4.1.3 Encoder

The encoder processes an action sequence(s) α and generates an individual latent representation $\boldsymbol{z} \in \mathbb{R}^M$ as

$$\boldsymbol{z} = \mathrm{ENC}(\alpha, \phi; \Theta_{\mathrm{ENC}}), \tag{2}$$

where ϕ is a problem in condition A, Θ_{ENC} represents the encoder's weights, and M is the dimensionality of the individual latent representation. The encoder architecture is task-condition-specific and designed for condition A.

4.1.4 Decoder

The decoder receives the individual latent representation $\boldsymbol{z}$ and generates the task solver's weights as

$$\Theta_{\mathrm{TS}} = \mathrm{DEC}(\boldsymbol{z}; \Theta_{\mathrm{DEC}}), \tag{3}$$

where Θ_{DEC} represents the decoder's weights. Since the decoder determines the task solver's weights, it functions as a hypernetwork [20, 22].

4.1.5 Training objective

Although conditions A and B differ, an individual's decision-making system remains consistent across task conditions. We model this using the individual latent representation $\boldsymbol{z}$, linking it to the task solver via the encoder and decoder. For training, we use behavioural dataset $\{\mathcal{A}_n, \mathcal{B}_n\}_{n \in \mathcal{P}}$ from a individual pool $\mathcal{P}$.

Let α be an action sequence representing individual n's behaviour on the source task condition, i.e., $(\alpha, \phi) \in \mathcal{A}_n$, $n \in \mathcal{P}$. The individual latent representation is derived by $\boldsymbol{z} = \mathrm{ENC}(\alpha, \phi; \Theta_{\mathrm{ENC}})$. The weights of the task solver are then given by $\Theta_{\mathrm{TS}} = \mathrm{DEC}(\boldsymbol{z}; \Theta_{\mathrm{DEC}})$. Subsequently, the task solver, with the given weights, predicts an action sequence for condition B as $\hat{\beta} = \mathrm{TS}(\psi; \Theta_{\mathrm{TS}})$, where $(\beta, \psi) \in \mathcal{B}_n$. We then measure the prediction error between $\hat{\beta}$ and β as:

$$L_p(\alpha, \phi, \beta, \psi, \Theta_{\mathrm{ENC}}, \Theta_{\mathrm{DEC}}) = O(\beta, \hat{\beta}), \tag{4}$$

where β is an action sequence in $\mathcal{B}_n$ recorded along with the problem ψ, and $O(\cdot, \cdot)$ is a suitable loss function (e.g., likelihood-based loss for probabilistic outputs). Using the datasets containing the behaviour of the individual pool $\mathcal{P}$, the weights of the encoder and decoders, Θ_{ENC} and Θ_{DEC}, are optimized by minimizing the total loss:

$$L(\Theta_{\mathrm{ENC}}, \Theta_{\mathrm{DEC}}) = \frac{1}{|\mathcal{P}|} \sum_{n \in \mathcal{P}} \frac{1}{|\mathcal{A}_n|} \sum_{(\alpha, \phi) \in \mathcal{A}_n} \frac{1}{|\mathcal{B}_n|} \sum_{(\beta, \psi) \in \mathcal{B}_n} L_p(\alpha, \phi, \beta, \psi, \Theta_{\mathrm{ENC}}, \Theta_{\mathrm{DEC}}). \tag{5}$$

This section provides a general formulation of individuality transfer across two task conditions. For specific details on task architectures and loss functions, see Sections 4.2 and 4.3.

4.2 Experiment on MDP task

We validated our individuality transfer framework using two different decision-making tasks: the MDP task and the MNIST task. This section focuses on the MDP tasks, a dynamic multi-step decision-making task.

4.2.1 Task

At the beginning of each episode, an initial state-cue is presented to the participant. For human participants, the state-cue is represented by animal images

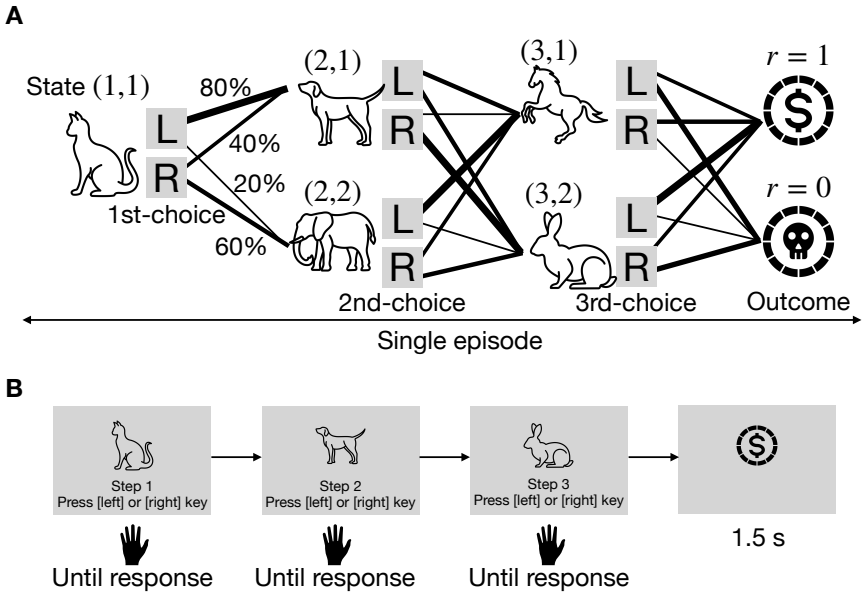


Figure 12: The 3-step MDP task. **A.** Tree diagram illustrating state-action transitions. **B.** Flow of a single episode in the behavioural experiment for human participants.

(Figure 12). For the cognitive model (Q-learning agent) and neural network-based model, the state-cue is represented numerically (e.g., $(2, 1)$ for the first task state in the second choice). The participant makes a binary decision (denoted as action C_1 or C_2) for each step. In the human experiment, these actions correspond to pressing the left or right cursor key. With a certain probability (either 0.8/0.2 or 0.6/0.4), known as the state-action transition probability, the participant transitions to one of two subsequent task states. This process repeats two times for the 2-step MDP and three times in the 3-step MDP. After the final step, the participant receives an outcome: either a reward ($r = 1$) or no reward ($r = 0$). For human participants, rewards were displayed as symbols, as shown in Figure 12. Each sequence from initial state-cue presentation to reward delivery constitutes an *episode*.

The state-action transition probability $T(s, a, s')$ from a task state s to a preceding state s' given an action a varies gradually across episodes. With probability p_{trans}, one of the transition probabilities switches to a new set chosen from $\{(0.8, 0.2), (0.2, 0.8), (0.6, 0.4), (0.4, 0.6)\}$. Consequently, participants must adjust their decision-making strategy in response to these shifts in transition probabilities to maintain reward maximization.

4.2.2 Behavioural data collection

We recruited 123 participants via Prolific. All participants provided their informed consent online. This study was approved by the Committee for Human Research at the Graduate School of Engineering, The University of Osaka, and compiled with the Declaration of Helsinki. Participants received a base compensation of £4 for completing the entire experiment, A performance-based bonus (£0 to £2, average: £1) was awarded based on rewards earned in the MDP task.

Each participant completed 3 sequences for each step condition (2-step and 3-step MDP tasks), with each sequence comprising 50 episodes. The order of the 2-step and 3-step MDP tasks was randomized across sequences. State-cue assignment (animal images) were randomly determined for each sequence. Participants took a mandatory break ($\geq$1 minute) between sequences.

To ensure data quality, we applied exclusion criteria based on average reward, action bias, and response time. Thresholds for these metrics were systematically determined using the interquartile range method on statistics from the initial dataset. Participants were removed from the analysis entirely if their data from any single block fell outside these established ranges. This procedure led to the exclusion of one participant for low average reward (below 0.387 for the 2-step MDP and 0.382 for the 3-step MDP), 23 participants for excessive action bias (outside the 26.3–73.3% range), and 18 for outlier response times (outside the 0.260–1.983 s range). In total, 42 participants (approximately 34%) were excluded, resulting in a final sample of 81 participants for analysis.

4.2.3 Cognitive model: Q-learning

To model decision-making in the MDP task, we employed a Q-learning agent [46]. At each step t, the agent was presented with the current task state s_t and selected an action a_t. The agent maintained Q-values, denoted as $Q(s, a)$, for all state-action pairs, where s was a state of the set of all possible task states $\mathcal{S}$ and a was an action of the set of available actions in that state $\mathcal{C}_s$. The probability of selecting action a was determined by a softmax policy:

$$\pi(a) = \frac{\exp(q_{\mathrm{it}} Q(s_t, a))}{\sum_{a' \in \mathcal{C}_{s_t}} \exp(q_{\mathrm{it}} Q(s_t, a'))}, \tag{6}$$

where $q_{\mathrm{it}} > 0$ was a parameter called the inverse temperature or reward sensitivity, controlling the balance between exploration and exploitation.

After selecting action a_t, the agent received an outcome $r_t \in \{0, 1\}$ and transitioned to a new state s_{t+1}. The Q-value for the selected action was updated by

$$Q(s_t, a_t) \leftarrow (1 - q_{\mathrm{lr}}) Q(s_t, a_t) + q_{\mathrm{lr}} (r_t + q_{\mathrm{dr}} \max_{a \in \mathcal{C}_{s_{t+1}}} Q(s_{t+1}, a)), \tag{7}$$

where $q_{\mathrm{lr}} \in (0, 1)$ was the learning rate, determining how much newly acquired information replaced existing knowledge, and $q_{\mathrm{dr}} \in (0, 1)$ was the discount rate, governing the extent to which future rewards influenced current decision. The Q-values are initialized as q_{init} before an agent starts first episode.

4.2.4 EIDT model

This section describes the specific models used for individuality transfer in the MDP task.

Data representation Since MDP tasks involve sequential decision-making, each action sequence consists of multiple actions within a single session. In our experiment, each participant completed L trials per session, with $L = 100$ for the 2-step MDP and $L = 150$ for the 3-step MDP. The action sequence is represented as $[(s_1, a_1, r_1), \ldots, (s_L, a_L, r_L)]$, where, s_t denotes the task state at trial t, $a_t \in C$ represents the action selected from the set $C \equiv \{C_k\}_{k=1}^{K}$ (with $K = 2$ in our task), and $r_t \in \{0, 1\}$ indicates whether a reward was received. In the M-step MDP described in Section 4.2, each task state is represented as (m, c_m), where m denotes the current step within the episode ($m \in \{1, \ldots, M\}$) and c_m corresponds to the cue presented to the participant. The action sequence, denoted as α or β, consists of a sequence of selected actions $(a_1, \ldots, a_L)$, while a problem, denoted as ϕ or ψ, is represented as $((s_1, \ldots, s_L), (r_1, \ldots, r_L))$.

Task solver Before describing the encoder and decoder, we define the architecture of the task solver, which generates actions for the M-step MDP task. The task solver is implemented using a gated recurrent unit (GRU) [7] with Q cells, where $Q = 4$ for the 2-step task and $Q = 8$ for the 3-step task. At time-step t, the GRU takes as input the previous hidden state $\boldsymbol{h}_{t-1} \in \mathbb{R}^Q$, the previous task state s_{t-1}, the previous action a_{t-1}, the previous reward r_{t-1}, and the current task state s_t. It then updates the hidden state as

$$\boldsymbol{h}_t = \mathrm{GRU}(s_{t-1}, a_{t-1}, r_{t-1}, s_t, \boldsymbol{h}_{t-1}; \Phi), \tag{8}$$

where Φ represents the GRU's weights. The updated hidden state is then used to predict the probability of selecting each action through a fully-connected feed forward layer:

$$\boldsymbol{v}_t = \boldsymbol{W}\boldsymbol{h}_t, \tag{9}$$

where $\boldsymbol{v}_t$ represents the logit scores for each action (unnormalised probabilities), and $\boldsymbol{W} \in \mathbb{R}^{K \times Q}$ is the weight matrix. The probabilities of each action are computed using a softmax layer:

$$\pi(a_t = C_k) = \frac{e^{[\boldsymbol{v}_t]_k}}{\sum_{k'=1,\ldots,K} e^{[\boldsymbol{v}_t]_{k'}}}, \tag{10}$$

where $\pi(a_t = C_k)$ represents the probability of selecting action C_k at time t, and $[\boldsymbol{v}_t]_i$ denotes the i-th element of $\boldsymbol{v}_t$.

For input encoding, we used a 1-of-K scheme. The step of the MDP task is encoded as $[1, 0, 0]$ for step 1, $[0, 1, 0]$ for step 2, and $[0, 0, 1]$ for step 3. Each task state s_m is represented as $[1, 0]$ or $[0, 1]$ to distinguish the two state cues at each step. The participant's action is encoded as C_1: $[1, 0]$ or C_2: $[0, 1]$, while the reward is represented as 0: $[1, 0]$ or 1: $[0, 1]$. These encodings are concatenated to form input sequences.

The task solver $\mathrm{TS}(\psi; \Theta_{\mathrm{TS}})$ generates a sequence of predicted action probabilities $\{(\pi(a_t = C_1), \ldots, \pi(a_t = C_K))\}_{t=1}^{L}$, using the GRU, the fully-connected layer $\boldsymbol{W}$, and the softmax layer. The problem ψ defines the MDP environment, specifying state transitions and reward outcomes in response to selected action.

To evaluate prediction accuracy, the loss function $O(\beta, \hat{\beta})$, defined in (4), compares human-performed action $\{\beta, \psi\}$ with those predicted by the task solver, $\{\hat{\beta}, \psi\}$. Notably, the problem ψ is not executed with the task solver; instead, the task solver predicts action probabilities based on the same task state and reward history as in the human behavioural data.

Encoder and decoder The encoder $\mathrm{ENC}(\alpha, \phi; \Theta_{\mathrm{ENC}})$ extracts an individual latent representation $\boldsymbol{z}$ from a sequence of actions α corresponding to a given environment ϕ. The first module of the encoder is a GRU, similar to the task solver, with $R = 32$ cells. The final hidden state $\boldsymbol{h}_L \in \mathbb{R}^R$ serves as the basis for computing the individual latent representation $[\boldsymbol{z} \in \mathbb{R}^M$ using a fully-connected feed-forward network with four layers $d(\cdot)$ as $\boldsymbol{z} = d(\boldsymbol{h}_L)$.

The decoder takes the individual latent representation $\boldsymbol{z}$ as input and generates the weights for the task solver by $\Theta_{\mathrm{TS}} = \mathrm{DEC}(\boldsymbol{z}; \Theta_{\mathrm{DEC}})$. The decoder is implemented as a single-layer linear network.

4.3 Experiment on MNIST task

This section describes the specific models used for individuality transfer in handwritten digit recognition (MNIST) task.

4.3.1 Task

The dataset used in this experiment was originally collected and published by Rafiei et al. [34]. In this task, participants were presented with a stimulus image depicting a handwritten digit and were required to respond by pressing the corresponding number key, as illustrated Figure 1. For further details regarding the task design and data collection, refer to [34].

4.3.2 EIDT model

Data representation An action sequence, denoted as α or β, consists of a single action a and its corresponding response time b. The associated problem, represented as ϕ or ψ, corresponds to a stimulus image. The action a is selected from a set $\{C_1, \ldots C_K\}$. Since the task involves recognizing digits ranging from 0 to 9, the number of possible actions is $K = 10$. The stimulus image, ϕ or ψ, is an image of size $H \times W$. In this experiment, we adopted the same resolution as [34], setting $H = W = 227$.

Task solver The task solver for the handwritten digit recognition task is based on the model proposed by [34]. Their model consists of a CNN and an evidence accumulation module. However, since their model represents average human

behaviour and does not account for individuality differences, we replace the accumulation module with a GRU [6] to capture individuality. The CNN module processes the input image and produces an evidence vector $\boldsymbol{e} = \mathrm{CNN}(\psi)$, where $\boldsymbol{e} \in \mathbb{R}^K$ and $\mathrm{CNN}(\cdot)$ is based on the AlexNet architecture [26]. The weights of the CNN are sampled from a Bayesian neural network (BNN), introducing stochasticity in the output. This stochasticity enables the models to generate human-like, probabilistic decisions.

The stimulus image is fed into the CNN S times, generating S evidence distributions $\boldsymbol{e}_t \in \mathbb{R}^K$ at each time step $t = 0, \ldots, S-1$. In this study, we set $S = 16$ to match the maximum response time, as described later. Since the CNN weights are stochastically sampled from the BNN, the CNN's output varies even when the same image is input multiple times. To model individuality in decision-making, we introduce a GRU with Q cells ($Q = 4$ in our setup). The GRU receives as input the previous hidden state $\boldsymbol{h}_{t-1} \in \mathbb{R}^Q$ and the current evidence $\boldsymbol{e}_t$, updating its hidden state as

$$\boldsymbol{h}_t = \mathrm{GRU}(\boldsymbol{e}_t, \boldsymbol{h}_{t-1}; \Phi), \tag{11}$$

where Φ represents the GRU's network weights. The updated hidden state is passed through a dense layer (as defined in (9)) and a softmax layer (as defined in (10)) to generate the probability distribution over possible digit classifications $[P_t(C_1), \ldots, P_t(C_K)]$ at each time step t.

To evaluate the prediction error, we compare the action sequences generated by human participants $\{\beta, \psi\}$ with those predicted by the task solver $\{\hat{\beta}, \psi\}$, incorporating response times into these analysis. The actual response time b is converted into an integer time step $\tilde{t}$ using the formula: $\tilde{t} = \mathrm{round}(10b)$. For example, a response time of $b = 0.765$ s is converted to $\tilde{t} = 8$. The likelihood of observed decision is then calculated as $P_{\tilde{t}}(a)$, where a is the actual digit chosen by the participant.

In this task solver, the CNN (driven by BNN) models a visual processing system, while the RNN represents the decision-making system. We assume that the visual system (implemented by CNN and BNN) is shared across all individuals, whereas the decision-making system (implemented by RNN) captures individual differences. Based on this assumption, the CNN and BNN are pretrained using the MNIST dataset [26], and their weight distributions are fixed across individuals. The pretraining procedure followed the original methodology [34].

Encoder and decoder Since each action sequence contains only a single action, it does not form a true "sequence." This makes it challenging to extract individuality from a single data point. To address this, the encoder takes a set of single action sequences as input rather than a single sequence. Specifically, the encoder $\mathrm{ENC}(\{\alpha_u, \phi_u\}_{u=1}^U; \Theta_{\mathrm{ENC}})$ extracts the individual latent representation $\boldsymbol{z}$ from U sets of stimulus images ϕ_u and their corresponding responses α_u, where $u = 1, \ldots, U$. Here, ϕ_u represent the stimulus presented in the u-th trial, and α_u represents the corresponding response. The number of action sequences U corresponds to the number of samples available for each individual in the

dataset. Since the outputs for these action sequences are just averaged, U can be adjusted flexibly.

The encoder architecture consists of a single CNN module, a single GRU, and a fully-connected feed-forward network. The CNN module is identical to the one used in the task solver. Given an input ϕ_u, let $\boldsymbol{e}_{t,u}$ represent the evidence output from the CNN at time step t. The GRU, which consists R cells ($R = 16$ in our setup), updates its hidden state based on the previous state, the current CNN evidence, and an encoding of the response action by

$$\boldsymbol{h}_{t,u} = \mathrm{GRU}(\boldsymbol{e}_{t,u}, k(a, t, \tilde{t}), \boldsymbol{h}_{t-1,u}; \Psi), \tag{12}$$

where Ψ represents the network weights. The function $k(a, t, \tilde{t})$ outputs the one-hot encoded action a if $t = \tilde{t}$, and zeros otherwise. The value $\tilde{t}$ represents the converted response time, obtained from the original response time b in the action sequence α_u. After processing all U sequences, the final hidden states are averaged across sequences: $\boldsymbol{h} = \frac{1}{U}\sum_{u=1}^{U} \boldsymbol{h}_{L,u}$. The individual latent representation is then computed as $\boldsymbol{z} = d(\boldsymbol{h})$, where $d(\cdot)$ represents a single-layer fully-connected feed-forward network. The decoder, implemented as a single linear layer, takes the individual latent representation $\boldsymbol{z}$ as input and outputs the weights for the task solver.

5 Data availability

The behavioural data for the MDP task has been made publicly available at `https://github.com/hgshrs/indiv_trans`

6 Code availability

All code and trained models have been made publicly available at `https://github.com/hgshrs/indiv_trans`

References

[1] N. J. Boogert, J. R. Madden, J. Morand-Ferron, and A. Thornton. Measuring and understanding individual differences in cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1756):20170280, 9 2018.

[2] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in beta-VAE. *arXiv preprint*, (arXiv:1804.03599), 4 2018.

[3] J. R. Busemeyer and J. C. Stout. A contribution of cognitive decision models to clinical assessment: Decomposing performance on the Bechara gambling task. *Psychological Assessment*, 14(3):253–262, 2002.

[4] M. Canizo, I. Triguero, A. Conde, and E. Onieva. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing*, 363:246–260, 10 2019.

[5] J. B. Carroll and S. E. Maxwell. Individual differences in cognitive abilities. *Annual Review of Psychology*, 30(1):603–640, 1 1979.

[6] Y.-A. Cheng, I. Felipe Rodriguez, S. Chen, K. Kar, T. Watanabe, and T. Serre. RTify: Aligning deep neural networks with human behavioral decisions. In *Advances in Neural Information Processing Systems*, 2024.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[8] A. G. E. Collins and M. J. Frank. How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, 35(7):1024–1035, 4 2012.

[9] N. D. Daw, S. J. Gershman, B. Seymour, P. Dayan, and R. J. Dolan. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 3 2011.

[10] A. Dezfouli, H. Ashtiani, O. Ghattas, R. Nock, P. Dayan, and S. O. Cheng. Disentangled behavioral representations. In *Advances in Neural Information Processing Systems*, 2019.

[11] A. Dezfouli, K. Griffiths, F. Ramos, P. Dayan, and B. W. Balleine. Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, 15(6):e1006903, 6 2019.

[12] K. D. Duncan and D. Shohamy. Memory states influence value-based decisions. *Journal of Experimental Psychology: General*, 145(11):1420–1426, 11 2016.

[13] M. K. Eckstein, S. L. Master, L. Xia, R. E. Dahl, L. Wilbrecht, and A. G. Collins. The interpretation of computational model parameters depends on the context. *eLife*, 11:75474, 11 2022.

[14] M. K. Eckstein, C. Summerfield, N. D. Daw, and K. J. Miller. Predictive and interpretable: combining artificial neural networks and classic cognitive models to understand human learning and decision making. In *Proceedings of the 45th Annual Conference of the Cognitive Science Society*, pages 928–935, 2023.

[15] T. Fel, I. Felipe, D. Linsley, and T. Serre. Harmonizing the object recognition strategies of deep neural networks with humans. In *Advances in Neural Information Processing Systems*, 2022.

[16] M. Fintz, M. Osadchy, and U. Hertz. Using deep learning to predict human decisions and using cognitive models to explain deep learning models. *Scientific Reports*, 12(1):4736, 3 2022.

[17] M. J. Frank, B. B. Doll, J. Oas-Terpstra, and F. Moreno. Prefrontal and striatal dopaminergic genes predict individual differences in exploration and exploitation. *Nature Neuroscience*, 12(8):1062–1068, 8 2009.

[18] Y. Ger, E. Nachmani, L. Wolf, and N. Shahar. Harnessing the flexibility of neural networks to predict dynamic theoretical parameters underlying human choice behavior. *PLOS Computational Biology*, 20(1):e1011678, 1 2024.

[19] Y. Ger, M. Shahar, and N. Shahar. Using recurrent neural network to estimate irreducible stochasticity in human choice behavior. *eLife*, 13, 9 2024.

[20] D. Ha, A. Dai, and Q. V. Le. Hypernetworks. *arXiv preprint*, (arXiv:1609.09106), 9 2016.

[21] K. Kar, J. Kubilius, K. Schmidt, E. B. Issa, and J. J. DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*, 22(6):974–983, 6 2019.

[22] T. Karaletsos, P. Dayan, and Z. Ghahramani. Probabilistic meta-representations of neural networks. *arXiv preprint*, (arXiv:1810.00555), 10 2018.

[23] K. Katahira. The relation between reinforcement learning parameters and the influence of reinforcement history on choice behavior. *Journal of Mathematical Psychology*, 66:59–69, 6 2015.

[24] M. Koivisto, H. Railo, A. Revonsuo, S. Vanni, and N. Salminen-Vaparanta. Recurrent processing in V1/V2 contributes to categorization of natural scenes. *The Journal of Neuroscience*, 31(7):2488–2492, 2 2011.

[25] N. Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1(1):417–446, 11 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[27] W.-H. Lin, A. M. Clarke, K. S. Pilz, M. H. Herzog, and M. Kunchulia. Intact reinforcement learning in healthy ageing. *bioRXiv*, 5 2023.

[28] K. J. Miller, M. Eckstein, M. M. Botvinick, and Z. Kurth-Nelson. Cognitive model discovery via disentangled RNNs. In *Advances in Neural Information Processing Systems*, pages 61377–61394, 2023.

[29] D. J. Navarro, T. L. Griffiths, M. Steyvers, and M. D. Lee. Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50(2):101–122, 4 2006.

[30] J. P. O'Doherty, A. Hampton, and H. Kim. Model-based fMRI and its application to reward learning and decision making. *Annals of the New York Academy of Sciences*, 1104(1):35–53, 5 2007.

[31] M. L. Pedersen, M. J. Frank, and G. Biele. The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic Bulletin and Review*, 24(4):1234–1251, 12 2017.

[32] J. C. Peterson, D. D. Bourgin, M. Agrawal, D. Reichman, and T. L. Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214, 6 2021.

[33] S. T. Radev, U. K. Mertens, A. Voss, L. Ardizzone, and U. Kothe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 4 2022.

[34] F. Rafiei, M. Shekhar, and D. Rahnev. The neural network RTNet exhibits the signatures of human perceptual decision-making. *Nature Human Behaviour*, 8(9):1752–1770, 7 2024.

[35] R. Rajalingham, E. B. Issa, P. Bashivan, K. Kar, K. Schmidt, and J. J. DiCarlo. Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *The Journal of Neuroscience*, 38(33):7255–7269, 8 2018.

[36] R. Ratcliff and G. McKoon. The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20(4):873–922, 4 2008.

[37] M. Rmus, T.-F. Pan, L. Xia, and A. G. E. Collins. Artificial neural networks for model identification and parameter estimation in computational cognitive models. *PLOS Computational Biology*, 20(5):e1012119, 5 2024.

[38] D. E. Rumelhart and J. L. McClelland. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pages 318–362. MIT Press, 1987.

[39] R. Schaeffer, M. Khona, L. Meshulam, and I. Fiete. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. In *Advances in Neural Information Processing Systems*, 2020.

[40] S. Sevy, K. E. Burdick, H. Visweswaraiah, S. Abdelmessih, M. Lukin, E. Yechiam, and A. Bechara. Iowa gambling task in schizophrenia: A review and new data in patients with schizophrenia and co-occurring cannabis use disorders. *Schizophrenia Research*, 92(1-3):74–84, 5 2007.

[41] W. Shengli. Is human digital twin possible? *Computer Methods and Programs in Biomedicine Update*, 1:100014, 1 2021.

[42] H. Siegelmann and E. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 2 1995.

[43] H. F. Song, G. R. Yang, and X.-J. Wang. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *eLife*, 6, 1 2017.

[44] M. Song, Y. Niv, and M. Bo Cai. Using recurrent neural networks to understand human reward learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, number 43, pages 1388–1394, 2021.

[45] C. J. Spoerer, T. C. Kietzmann, J. Mehrer, I. Charest, and N. Kriegeskorte. Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLOS Computational Biology*, 16(10):e1008215, 10 2020.

[46] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 1998.

[47] H. Tang, M. Schrimpf, W. Lotter, C. Moerman, A. Paredes, J. Ortega Caro, W. Hardesty, D. Cox, and G. Kreiman. Recurrent computations for visual pattern completion. *Proceedings of the National Academy of Sciences*, 115(35):8835–8840, 8 2018.

[48] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein autoencoders. *arXiv preprint*, (arXiv:1711.01558), 11 2017.

[49] D. Tuzsus, A. Brands, I. Pappas, and J. Peters. Exploration–exploitation mechanisms in recurrent neural networks and human learners in restless bandit problems. *Computational Brain & Behavior*, 5 2024.

[50] R. van den Bos, J. Homberg, and L. de Visser. A critical review of sex differences in decision-making tasks: Focus on the Iowa gambling task. *Behavioural Brain Research*, 238(1):95–108, 2 2013.

[51] D. Vickers. Evidence for an accumulator model of psychophysical discrimination. *Ergonomics*, 13(1):37–58, 4 2007.

[52] E.-J. Wagenmakers and S. Brown. On the linear relation between the mean and the standard deviation of a response time distribution. *Psychological Review*, 114(3):830–841, 7 2007.

[53] R. Wang, Y. Shen, P. Tino, A. Welchman, and Z. Kourtzi. Learning predictive statistics: strategies and brain mechanisms. *The Journal of Neuroscience*, 37(35):0144–17, 2017.

[54] R. C. Wilson and A. G. Collins. Ten simple rules for the computational modeling of behavioral data. *eLife*, 8:1–33, 2019.

[55] D. L. K. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 3 2016.

[56] G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, 2 2019.

[57] E. Yechiam, J. R. Busemeyer, J. C. Stout, and A. Bechara. Using cognitive models to map relations between neuropsychological disorders and human decision-making deficits. *Psychological Science*, 16(12):973–978, 12 2005.

Acknowledgements

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI, grant number 22H05163 and 24K15047, and Japan Science and Technology Agency (JST) Advanced International Collaborative Research Program (AdCORP), grant number JPMJKB2307. We appreciate Kaede Hashiguchi and Yuichi Tanaka, Graduate School of Engineering, The University of Osaka, who gave useful comments for this research.

Author contributions

H.H. designed and performed the research, collected a part of the data, analyzed the data, and drafted and edited the paper.

Competing interests

The authors declare no competing interests.